

GammaLib - Support #1939

custom model skymap

03/01/2017 10:12 PM - Kelley-Hoskins Nathan

Status:	New	Start date:	03/01/2017
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>I've got a GObservations from running ctlike, but after digging around in ctools, it seems like ctmodel doesn't accept GObservations as an input, only file paths, so I'm trying to get the model values in my own python code. My problem is I'm not sure how to properly get the model values.</p> <pre># I've got a GObservations from ctlike: like = ctools.ctlike(obs) # setup ctlike like.run() fitobs = like.obs() # and I've got a list of GSkyDirs for positions I want to make the map of skydirs = [...] # I think I need to setup an event bin to give to model.eval(), but I'm not sure. ebin = gammalib.GCTAEventBin() # energy bin from 0.5 to 1.5 TeV? ebin.energy(gammalib.GEnergy(1.0, 'TeV')) # linear mean of energy range ebin.ewidth(gammalib.GEnergy(0.5, 'TeV')) # linear energy bin width ebin.solidangle(0.01) # dummy value ebin.weight(1.0) # create an empty list for holding the model values, one per skydir mcounts = numpy.zeros(len(skydirs)) # loop over each sky direction to evaluate the models at for i, pixdir in enumerate(skydirs) : # move the event bin around for each skydir ebin.dir(pixdir) for ob in fitobs : ebin.time(ob.time()) ebin.ontime(ob.ontime()) for model in fitobs.models() : # for each combination of model + observation, get the model's predicted number of counts modcounts[ipix] += model.eval(ebin, ob, False)</pre> <p>Does this code look sane? Does this set ebin.time() and ontime() properly?</p>			

History

#1 - 03/01/2017 10:17 PM - Knödseder Jürgen

You mean

```
# I've got a GObservations from ctlike:
like = ctools.ctlike(obs)
```

```
# setup ctlike
like.run()
fitobs = like.obs()
# Now into ctmodel
model = ctools.ctmodel(fitobs)
```

does not work? Or do you need something else?

#2 - 03/02/2017 10:37 AM - Kelley-Hoskins Nathan

I had forgotten about `ctools.ctmodel(fitobs)` until now, but I had thought `ctmodel` outputs a plot, rather than (as I see now) a counts cube. Does the counts cube have an option for a list of specific pointings? I've been making plots with healpy bins rather than cartesian bins.

#3 - 03/02/2017 04:02 PM - Kelley-Hoskins Nathan

I've got this code now:

```
modcounts = numpy.zeros( len(fovpix) )
ebin = gammalib.GCTAEventBin()
ebin.energy( ( en_max + en_min ) / 2.0 )
ebin.ewidth( en_max - en_min )
ebin.solidangle( veripy.healpy_res2pixel_area(res) )
ebin.weight( 1.0 )
for ipix, pixdir in enumerate( fovskydir ) :
    for ob in like.obs() :
        ebin.dir( ob.pointing().instdir( pixdir ) )
        ebin.ontime( ob.ontime() )
        ebin.time( ob.gti().tstart() )
        for model in obs.models() :
            modcounts[ipix] += model.eval( ebin, ob, False )
```

which doesn't throw errors, but the model values in `modcounts` are one or two orders of magnitude lower than the counts map, so I think something's still wrong with this. And it's pretty slow too (~25 pixels / min).

After some talking with Gernot about sky map binning, I'm going to see if I can switch from healpy bins to just using `ctbin`, `ctmodel`, and `aply`.

#4 - 06/07/2017 12:47 AM - Knödseder Jürgen

You cannot compare a model to counts. A model (except for a background model) is living in physical coordinates, the counts are measured coordinates. You need to apply the IRF on the model to compare with counts. Why don't you simply take the model cube produced by `ctmodel`? This

can be directly compared to the counts cube.